IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:)	
)	
Robert W. SCHEIFLER et al.)	Group Art Unit: 2152
)	
Continuation of U.S. Patent)	Examiner: F. Farahi
Application No. 9/044,826)	
)	
Filed: **AUGUST 17, 2001**)	
)	
For: METHOD AND SYSTEM FOR)	
FACILITATING ACCESS TO A)	
LOOKUP SERVICE)	

Assistant Commissioner for Patents
Washington, DC 20231

Sir:

PRELIMINARY AMENDMENT

Prior to the examination of the above application, please amend this application as follows:

IN THE SPECIFICATION:

Please amend the specification as follows:

Please replace the following paragraphs on pages 1 through 4 beginning at page 1, line 6, with the following new paragraphs:

Provisional U.S. Patent Application No. 60/076,048, entitled "Distributed Computing System," filed on February 26, 1998.

U.S. Patent Application No. 09/044,923, entitled "Method and System for Leasing Storage," bearing attorney docket no. 06502.0011-01000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,838, entitled "Method, Apparatus, and Product for Leasing of Delegation Certificates in a Distributed System," bearing attorney docket no. 06502.0011-02000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,834, entitled "Method, Apparatus and Product for Leasing of Group Membership in a Distributed System," bearing attorney docket no. 06502.0011-03000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,916, entitled "Leasing for Failure Detection," bearing attorney docket no. 06502.0011-04000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,933, entitled "Method for Transporting Behavior in Event Based System," bearing attorney docket no. 06502.0054-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,919, entitled "Deferred Reconstruction of Objects and Remote Loading for Event Notification in a Distributed System," bearing attorney docket no. 06502.0062-01000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,938, entitled "Methods and Apparatus for Remote Method Invocation," bearing attorney docket no. 06502.0102-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/045,652, entitled "Method and System for Deterministic Hashes to Identify Remote Methods," bearing attorney docket no. 06502.0103-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,790, entitled "Method and Apparatus for Determining Status of Remote Objects in a Distributed System," bearing attorney docket no. 06502.0104-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,930, entitled "Downloadable Smart Proxies for Performing Processing Associated with a Remote Procedure Call in a Distributed System," bearing attorney docket no. 06502.0105-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,917, entitled "Suspension and Continuation of Remote Methods," bearing attorney docket no. 06502.0106-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,835, entitled "Method and System for Multi-entry and Multi-Template Matching in a Database," bearing attorney docket no. 06502.0107-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,839, entitled "Method and System for In-Place Modifications in a Database," bearing attorney docket no. 06502.0108-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,945, entitled "Method and System for Typesafe Attribute Matching in a Database," bearing attorney docket no. 06502.0109-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,931, entitled "Dynamic Lookup Service in a Distributed System," bearing attorney docket no. 06502.0110-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,939, entitled "Apparatus and Method for Providing Downloadable Code for Use in Communicating with a Device in a Distributed System," bearing attorney docket no. 06502.0112-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,932, entitled "Apparatus and Method for Dynamically Verifying Information in a Distributed System," bearing attorney docket no. 06502.0114-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/030,840, entitled "Method and Apparatus for Dynamic Distributed Computing Over a Network," bearing attorney docket no. 06502.0115-00000, and filed on February 26, 1988.

U.S. Patent Application No. 09/044,936, entitled "An Interactive Design Tool for Persistent Shared Memory Spaces," bearing attorney docket no. 06502.0116-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,934, entitled "Polymorphic Token-Based Control," bearing attorney docket no. 06502.0117-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,915, entitled "Stack-Based Access Control," bearing attorney docket no. 06502.0118-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,944, entitled "Stack-Based Security Requirements," bearing attorney docket no. 06502.0119-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,837, entitled "Pre-Method Designation of Security Requirements," bearing attorney docket no. 06502.0120-00000, and filed on the same date herewith.

On page 26, replace the paragraph on line 7 through line 16 with the following new paragraph:

The lookup service may contain a subset of all services available in the network, referred to as a "Djinn" as described in copending U.S. Patent Application Serial No. 09/044,931, entitled "Dynamical Lookup Service in a Distributed System," assigned to a common assignee, filed on even date herewith, which has been previously incorporated by reference. A "Djinn" refers to a logical grouping of one or more of the services or resources that are provided by a network. Devices connected to the network may either dynamically add themselves to the Djinn or dynamically remove themselves from the Djinn. When added, a device provides zero or more of its services to Djinn and may utilize all of the services currently provided by the Djinn. The services provided by the Djinn are defined by the lookup service, which provides a common way to both find and utilize the services for the Djinn.

On page 28, replace the paragraph on line 1 through line 2 with the following new paragraph:

09/044,839, entitled "Method and System for In-Place Modifications In A Database," previously incorporated herein.

On page 29, replace the paragraph on line 17 through line 25 with the following new paragraph:

Programs (including other services) that need a particular type of service can use the lookup service 400 to find a stub that can be used to access the service. A match can be made based on the type of service as well as the specific attributes attached to the service. For example, a client could search for a printer by requesting a stub type corresponding to the service desired or by requesting certain attributes such as a specific location or printing speed. In one implementation consistent with the present invention, attributes are stored as multi-entries, and a match on attributes can be made using multi-templates, as explained in co-pending U.S. Patent Application No. 09/044,835, entitled "Method and System for Multi-Entry and Multi-Template Matching In A Database," previously incorporated herein.

On page 30, replace the paragraph on line 2 through line 14 with the following new paragraph:

Referring back to FIG. 4, the stub 404 corresponding to a service is registered in the lookup service 400 and is used by the client computer 11(n) to access the service methods remotely. This stub 404 may also be a "smart proxy." A smart proxy, code within which a stub is embedded, helps the client more efficiently implement the stub and the method to be remotely invoked. A smart proxy often performs some local computation for efficiency before or after it actually calls the stub. For example, a smart proxy may contain code to cache information, so if a client requested it again, instead of going back to the server to get the information, it may have cached the answer and be able to return it quickly. If the situation called for it, a smart proxy might also transform the parameters received from the client into other types and then send the transformed types. The smart proxy concept is further explained in co-pending U.S. Patent Application No. 09/044,930, entitled "Downloadable Smart Proxies for Performing Processing Associated with a Remote Procedure Call in a Distributed System," assigned to a common assignee, filed on even date herewith, which is hereby incorporated by reference.

IN THE CLAIMS:

Please cancel claim 1 without prejudice and without forfeiting any of the claimed subject matter, and add new claims 28-61, as follows:

28.   (New)   A method in a data processing system for accessing network services associated with a lookup service, comprising the steps of:

receiving a request from a client by the lookup service for access to one of the network services, the client being remote with respect to the lookup service; and

returning a resource locator to the client from the lookup service so that the client may dynamically load executable code to facilitate access of the one network service.

29.   (New)   The method of claim 28, further comprising the step of:

using the returned resource locator to dynamically load executable code to facilitate access of the one network service.

30.   (New)   The method of claim 29, further comprising the step of:

accessing the network service by the client using the dynamically loaded executable code.

31.   (New)   The method of claim 28, wherein the step of returning a resource locator includes the step of:

returning stub information to the client.

32.   (New)   The method of claim 31, further comprising the step of:

using the resource locator in the client to dynamically load executable code for the stub.

33.   (New)   The method of claim 32, further comprising the step of:

accessing the network service by the client using the dynamically loaded

executable code.

34.   (New)   The method of claim 28, wherein the step of returning a resource

locator includes the step of:

returning smart proxy information to the client.

35.   (New)   The method of claim 34, further comprising:

using the resource locator in the client to dynamically load executable code for a

smart proxy.

36.   (New)   The method of claim 35, further comprising the step of:

accessing the network service by the client using the dynamically loaded

executable code.

37.   (New)   A method in a data processing system having a lookup service

with a plurality of services, the method comprising the steps of:

sending from a client to the lookup service a request to access one of the

services, the lookup service being remote with respect to the client; and

receiving, responsive to the request, by the client from the lookup service, a

resource locator to dynamically load executable code to facilitate access of the one

service.

38.   (New)   The method of claim 37, further comprising the step of:

using the received resource locator to dynamically load executable code to facilitate access of the one network service.

39.   (New)   The method of claim 38, further comprising the step of:

accessing the service by the client using the dynamically loaded executable code.

40.   (New)   The method of claim 37, wherein the step of receiving a resource locator comprises the step of:

receiving stub information; and

using the resource locator to dynamically load executable code for the stub.

41.   (New)   The method of claim 40, further comprising the step of:

accessing the network service by the client using the dynamically loaded executable code.

42.   (New)   The method of claim 37, wherein the step of receiving a resource locator comprises the step of:

receiving smart proxy information; and

using the resource locator to dynamically load executable code for a smart proxy.

43.   (New)   The method of claim 40, further comprising the step of:

accessing the network service by the client using the dynamically loaded

executable code.

44.   (New)   A distributed system with a plurality of network services,

comprising:

a server computer with a lookup service having a resource locator corresponding

to dynamically executable code for facilitating access to one of the network services;

and

a client computer with a program that sends a request to the lookup service for

the one network service, that receives the resource locator from the server computer,

and uses the resource locator to dynamically load executable code to facilitate access

of the one network service.

45.   (New)   A data processing system for accessing network services

associated with a lookup service, comprising:

means for receiving a request from a client by the lookup service for access to

one of the network services, the client being remote with respect to the lookup service;

and

means for returning a resource locator to the client from the lookup service such

that the client may dynamically load executable code to facilitate access of the one

network service.

-11-

46. (New) A computer-readable medium containing instructions for controlling a data processing system to perform a method for accessing network services associated with a lookup service, the method comprising the steps of:

receiving a request from a client by the lookup service for access to one of the network services, the client being remote with respect to the lookup service; and

returning a resource locator to the client from the lookup service so that the client may dynamically load executable code to facilitate access of the one network service.

47. (New) The computer-readable medium of claim 46, wherein the method further comprises the step of:

using the returned resource locator to dynamically load executable code to facilitate access of the one network service.

48. (New) The computer-readable medium of claim 47, wherein the method further comprises the step of:

accessing the network service by the client using the dynamically loaded executable code.

49. (New) The computer-readable medium of claim 46, wherein the step of a returning a resource locator comprises the step of:

returning stub information to the client.

50.   (New)   The computer-readable medium of claim 49, wherein the method further comprises the step of:

using the resource locator in the client to dynamically load executable code for the stub.

51.   (New)   The computer-readable medium of claim 50, wherein the method further comprises the step of:

accessing the network service by the client using the dynamically loaded executable code.

52.   (New)   The computer-readable medium of claim 46, wherein the step of returning a resource locator comprises the step of:

returning smart proxy information to the client.

53.   (New)   The computer-readable medium of claim 46, wherein the step of returning a resource locator comprises the step of:

using the resource locator in the client to dynamically load executable code for a smart proxy.

54.   (New)   The computer-readable medium of claim 53, wherein the method further comprises the step of:

accessing the network service by the client using the dynamically loaded executable code.

55. (New) A computer-readable medium containing instructions for controlling a data processing system to perform a method for accessing network services associated with a lookup service, the method comprising the steps of:

sending from a client to the lookup service a request to access one of the services, the lookup service being remote with respect to the client; and

receiving, responsive to the request, by the client from the lookup service, a resource locator to dynamically load executable code to facilitate access of the one service.

56. (New) The computer-readable medium of claim 55, wherein the method further comprises the step of:

using the received resource locator to dynamically load executable code to facilitate access of the one network service.

57. (New) The computer-readable medium of claim 56, wherein the method further comprises the step of:

accessing the network service by the client using the dynamically loaded executable code.

58. (New) The computer-readable medium of claim 55, wherein the step of receiving a resource locator includes the step of:

receiving stub information; and

using the resource locator to dynamically load executable code for the stub.

59.   (New)   The computer-readable medium of claim 58, wherein the method further comprises the step of:

accessing the network service by the client using the dynamically loaded executable code.

60.   (New)   The computer-readable medium of claim 55, wherein the step of receiving a resource locator includes the step of:

receiving smart proxy information; and

using the resource locator in the client to dynamically load executable code for a smart proxy.

61.   (New)   The computer-readable medium of claim 60, wherein the method further comprises the step of:

accessing the network service by the client using the dynamically loaded executable code.

REMARKS

Applicants has added the appropriate patent application serial numbers. Additionally, Applicants have canceled claim 1 without prejudice and without forfeiting any of the claimed subject matter, and added new claims 28-61.
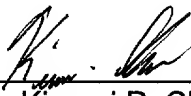
If there is any fee due in connection with the filing of this Preliminary

Amendment, please charge the fee to our Deposit Account No. 06-0916.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

Dated: _____August 17, 2001_____   By:_____

Kimani P. Clark
Reg. No. 45,969

APPENDIX

IN THE SPECIFICATION:

Please amend the specification as follows:

Please replace the following paragraphs on pages 1 through 4 beginning at page 1, line 6, with the following new paragraphs:

Provisional U.S. Patent Application No. 60/076,048, entitled "Distributed Computing System," filed on February 26, 1998.

U.S. Patent Application No. 09/044,923, entitled "Method and System for Leasing Storage," bearing attorney docket no. 06502.0011-01000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,838, entitled "Method, Apparatus, and Product for Leasing of Delegation Certificates in a Distributed System," bearing attorney docket no. 06502.0011-02000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,834, entitled "Method, Apparatus and Product for Leasing of Group Membership in a Distributed System," bearing attorney docket no. 06502.0011-03000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,916, entitled "Leasing for Failure Detection," bearing attorney docket no. 06502.0011-04000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,933, entitled "Method for Transporting Behavior in Event Based System," bearing attorney docket no. 06502.0054-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,919, entitled "Deferred Reconstruction of Objects and Remote Loading for Event Notification in a Distributed System," bearing attorney docket no. 06502.0062-01000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,938, entitled "Methods and Apparatus for Remote Method Invocation," bearing attorney docket no. 06502.0102-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/045,652, entitled "Method and System for Deterministic Hashes to Identify Remote Methods," bearing attorney docket no. 06502.0103-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,790, entitled "Method and Apparatus for Determining Status of Remote Objects in a Distributed System," bearing attorney docket no. 06502.0104-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,930, entitled "Downloadable Smart Proxies for Performing Processing Associated with a Remote Procedure Call in a Distributed System," bearing attorney docket no. 06502.0105-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,917, entitled "Suspension and Continuation of Remote Methods," bearing attorney docket no. 06502.0106-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,835, entitled "Method and System for Multi-Entry and Multi-Template Matching in a Database," bearing attorney docket no. 06502.0107-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,839, entitled "Method and System for In-Place Modifications in a Database," bearing attorney docket no. 06502.0108-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,945, entitled "Method and System for Typesafe Attribute Matching in a Database," bearing attorney docket no. 06502.0109-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,931, entitled "Dynamic Lookup Service in a Distributed System," bearing attorney docket no. 06502.0110-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,939, entitled "Apparatus and Method for Providing Downloadable Code for Use in Communicating with a Device in a Distributed System," bearing attorney docket no. 06502.0112-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,932, entitled "Apparatus and Method for Dynamically Verifying Information in a Distributed System," bearing attorney docket no. 06502.0114-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/030,840, entitled "Method and Apparatus for Dynamic Distributed Computing Over a Network," bearing attorney docket no. 06502.0115-00000, and filed on February 26, 1988.

U.S. Patent Application No. 09/044,936, entitled "An Interactive Design Tool for Persistent Shared Memory Spaces," bearing attorney docket no. 06502.0116-00000, and filed on the same date herewith.

U.S. Patent Application No. <u>09/044,934</u>, entitled "Polymorphic Token-Based Control," bearing attorney docket no. 06502.0117-00000, and filed on the same date herewith.

U.S. Patent Application No. <u>09/044,915</u>, entitled "Stack-Based Access Control," bearing attorney docket no. 06502.0118-00000, and filed on the same date herewith.

U.S. Patent Application No. <u>09/044,944</u>, entitled "Stack-Based Security Requirements," bearing attorney docket no. 06502.0119-00000, and filed on the same date herewith.

U.S. Patent Application No. <u>09/044,837</u>, entitled "Pre-Method Designation of Security Requirements," bearing attorney docket no. 06502.0120-00000, and filed on the same date herewith.

LAW OFFICES
FINNEGAN, HENDERSON,
FARABOW, GARRETT,
& DUNNER, L.L.P.
1300 I STREET, N. W.
WASHINGTON, DC 20005
202-408-4000

On page 26, replace the paragraph on line 7 through line 16 with the following new paragraph:

The lookup service may contain a subset of all services available in the network, referred to as a "Djinn" as described in copending U.S. Patent Application Serial No. 09/044,931, entitled "Dynamical Lookup Service in a Distributed System," assigned to a common assignee, filed on even date herewith, which has been previously incorporated by reference. A "Djinn" refers to a logical grouping of one or more of the services or resources that are provided by a network. Devices connected to the network may either dynamically add themselves to the Djinn or dynamically remove themselves from the Djinn. When added, a device provides zero or more of its services to Djinn and may utilize all of the services currently provided by the Djinn. The services provided by the Djinn are defined by the lookup service, which provides a common way to both find and utilize the services for the Djinn.

On page 28, replace the paragraph on line 1 through line 2 with the following new paragraph:

09/044,839, entitled "Method and System for In-Place Modifications In A Database," previously incorporated herein.

On page 29, replace the paragraph on line 17 through line 25 with the following new paragraph:

Programs (including other services) that need a particular type of service can use the lookup service 400 to find a stub that can be used to access the service. A match can be made based on the type of service as well as the specific attributes attached to the service. For example, a client could search for a printer by requesting a stub type corresponding to the service desired or by requesting certain attributes such as a specific location or printing speed. In one implementation consistent with the present invention, attributes are stored as multi-entries, and a match on attributes can be made using multi-templates, as explained in co-pending U.S. Patent Application No. 09/044,835, entitled "Method and System for Multi-Entry and Multi-Template Matching In A Database," previously incorporated herein.

On page 30, replace the paragraph on line 2 through line 14 with the following new paragraph:

Referring back to FIG. 4, the stub 404 corresponding to a service is registered in the lookup service 400 and is used by the client computer 11(n) to access the service methods remotely. This stub 404 may also be a "smart proxy." A smart proxy, code within which a stub is embedded, helps the client more efficiently implement the stub and the method to be remotely invoked. A smart proxy often performs some local computation for efficiency before or after it actually calls the stub. For example, a smart proxy may contain code to cache information, so if a client requested it again, instead of going back to the server to get the information, it may have cached the answer and be able to return it quickly. If the situation called for it, a smart proxy might also transform the parameters received from the client into other types and then send the transformed types. The smart proxy concept is further explained in co-pending U.S. Patent Application No. 09/044,930, entitled "Downloadable Smart Proxies for Performing Processing Associated with a Remote Procedure Call in a Distributed System," assigned to a common assignee, filed on even date herewith, which is hereby incorporated by reference.